

Họ tên: Nguyễn

Tiến Hùng

MSV:

B20DCCN297


- Apps

- catalog

- catalog/models.py


```
ShoppingCart > catalog > models.py > ...
1  from django.db import models
2  from django.contrib.auth.models import User
3  from db_mongo_connection import db
4
5  # Create your models here.
6  category_collection = db['Category']
7
8
9  class Category:
10     def __init__(self, name, slug, description, is_active=True):
11         self.name = name
12         self.slug = slug
13         self.description = description
14         self.is_active = is_active
15         self.created_at = None # You can set these values based on your re
16         self.updated_at = None
17
18     def __str__(self):
19         return self.name
20
```

- catalog/views.py

ShoppingCart > catalog >  views.py > ...


```
1  from django.shortcuts import render
2  from .models import category_collection, Category
3  from django.http import HttpResponse
4
5
6  # Create your views here.
7  def init(request):
8      records = [
9          {
10             "name": "Thiếu Nhi",
11             "slug": "thieu-nhi",
12             "description": "Sách thiếu nhi",
13         },
14         {
15             "name": "Khoa Học",
16             "slug": "khoa-hoc",
17             "description": "Sách khoa học"
18         },
19         {
20             "name": "Kinh Tế",
21             "slug": "kinh-te",
22             "description": "Sách kinh tế"
23         }
24     ]
25
26     category_collection.insert_many(records)
27
28     return HttpResponse("New category is added")
29
```

- catalog/urls.py

ShoppingCart > catalog >  urls.py > ...

```
1  from django.contrib import admin
2  from django.urls import path, include
3  from . import views
4
5  urlpatterns = [
6      path('init/', views.init, name='init_catalog')
7  ]
```

- product
 - product/models.py

ShoppingCart > product >  models.py > ...

```
1  from django.db import models
2  from db_mongo_connection import db
3  from catalog.models import Category
4
5  # Create your models here.
6  product_collection = db['Product']
7
8
9  class Product:
10     def __init__(self, name, author, slug, price, image, is_active=True,
11                 is_bestseller=False, quantity=0, description='', categories=None):
12         self.name = name
13         self.author = author
14         self.slug = slug
15         self.price = price
16         self.image = image
17         self.is_active = is_active
18         self.is_bestseller = is_bestseller
19         self.quantity = quantity
20         self.description = description
21         self.created_at = None # You can set these values based on your requirements
22         self.updated_at = None
23         self.categories = categories or []
24
25     def __str__(self):
26         return self.name
27
28
```

- **product/views.py**



```
ShoppingCart > product > 📁 views.py > 📄 add_product
1  from django.shortcuts import render
2  from .models import product_collection
3  from django.http import HttpResponse
4  from .utils import parse_product_from_mongo
5
6
7  # Create your views here.
8  def init(request):
9      records = [
10 >         { ...
27 >         { ...
46 >         { ...
67 >         { ...
84 >         { ...
101 >        { ...
113     ]
114     product_collection.insert_many(records)
115     return HttpResponse("New products is added")
116
117
118     def index(request):
119         return HttpResponse("<h1>Product App</h1>")
120
```

```

120
121
122 def add_product(request):
123     records = {
124         "name": "ProductTest",
125         "price": 20000,
126         "digital": True,
127         "image": None
128     }
129
130     product_collection.insert_one(records)
131     return HttpResponse("New product is added")
132
133
134 def get_all_product(request):
135     mongo_data = product_collection.find()
136
137     products = [parse_product_from_mongo(data) for data in mongo_data]
138     print(products)
139     return HttpResponse(products)
140

```

- `product/urls.py`

ShoppingCart > product >  urls.py > ...

```
1  from django.contrib import admin
2  from django.urls import path, include
3  from . import views
4
5  urlpatterns = [
6      path('', views.index, name='product_index'),
7      path('init/', views.init, name='init_products'),
8      path('add/', views.add_product, name='product_add'),
9      path('get_all/', views.get_all_product, name='product_get_all'),
10 ]
11
```

- cart
 - cart/models.py

ShoppingCart > cart > models.py > ...

```
1  from django.db import models
2  from django.contrib.auth.models import User
3
4
5  class Cart(models.Model):
6      id = models.AutoField(primary_key=True)
7      user = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
8      active = models.BooleanField(default=True)
9      created_at = models.DateTimeField(auto_now_add=True)
10     updated_at = models.DateTimeField(auto_now=True)
11
12     def add_to_cart(self, product_slug):
13         cart_item = CartItem.objects.get_or_create(product_slug=product_slug)
14         self.items.add(cart_item)
15         self.save()
16
17     def remove_from_cart(self, product_slug):
18         cart_item = CartItem.objects.get(product_slug=product_slug)
19         self.items.remove(cart_item)
20         self.save()
21
22     def remove_all_from_cart(self):
23         for item in self.items.all():
24             self.items.remove(item)
25         self.save()
26
27
```



```
27
28
29 # Create your models here.
30 class CartItem(models.Model):
31     id = models.AutoField(primary_key=True)
32     price = models.FloatField(default=0)
33     created_at = models.DateTimeField(auto_now_add=True)
34     updated_at = models.DateTimeField(auto_now=True)
35     product_slug = models.CharField(max_length=255, default='')
36     cart_id = models.ForeignKey(Cart, on_delete=models.CASCADE, null=
37     blank=True, related_name='cart_details')
38     quantity = models.IntegerField(null=False, default=0)
39
```

- **cart/views.py**

ShoppingCart > cart > views.py > addToCart

```
14
15 def addToCart(request, product_slug):
16     try:
17         product = product_collection.find({'slug': product_slug})[0]
18         cart = Cart.objects.first()
19         if not cart:
20             cart = Cart(
21                 )
22             cart.save()
23         obj, created = CartItem.objects.update_or_create(
24             cart_id_id=cart.id,
25             product_slug=product_slug,
26             defaults={
27                 'price': product['price'],
28             }
29         )
30         if not created:
31             obj.quantity += 1
32         else:
33             obj.quantity = 1
34         obj.save()
35         messages.success(request, "Thành công .")
36         return redirect(request.META.get('HTTP_REFERER', 'home'))
37
38     except Exception as e:
39         return HttpResponse(str(e))
40
```


ShoppingCart > cart > views.py > addToCart

```
40
41
42 def viewCart(request):
43     try:
44         cart = Cart.objects.first()
45         if not cart:
46             cart = Cart(
47             )
48             cart.save()
49
50         data = []
51
52         for cart_detail in cart.cart_detail.all():
53             product = product_collection.find({'slug': cart_detail.pr
54             product['_id'] = str(product['_id'])
55             data.append({
56                 'id': cart_detail.id,
57                 'price': cart_detail.price,
58                 'quantity': cart_detail.quantity,
59                 'product': product,
60                 'sum_price': cart_detail.price * cart_detail.quantity
61             })
62         # cartDetail = cart.cart_detail.all()
63         return render(request, "cart.html", get_context({"cart": data
64
65     except Exception as e:
66         return HttpResponse(str(e))
67
```

ShoppingCart > cart > views.py > addToCart


```
67
68
69 def changeQuantity(request):
70     try:
71         if request.method == 'GET':
72             raise Exception('Not support method')
73         cart_detail_id = request.POST.get('id')
74         type = request.POST.get('type')
75
76         cartDetail = CartItem.objects.get(pk=cart_detail_id)
77
78         if type == '+':
79             cartDetail.quantity += 1
80             cartDetail.save()
81
82         else:
83             if cartDetail.quantity == 1:
84                 cartDetail.delete()
85             else:
86                 cartDetail.quantity -= 1
87                 cartDetail.save()
88
89         messages.success(request, "Thành công .")
90         return redirect(request.META.get('HTTP_REFERER', 'car
91     except Exception as e:
92         return HttpResponse(str(e))
93
```

- cart/urls.py

ShoppingCart > cart >  urls.py > ...

```
1 from django.contrib import admin
2 from django.urls import path, include
3 from . import views
4
5 urlpatterns = [
6     path('add_to_cart/<str:product_slug>/', views.addToCart, name='add_to_cart')
7 ]
```

- app(checkout+search)
 - app/models.py

ShoppingCart > app >  models.py > ...

```
1 from django.db import models
2 from django.contrib.auth.models import User
3 from product.models import Product
4
5
6 class Customer(models.Model):
7     user = models.OneToOneField(User, null=True, blank=False,
8                                 on_delete=models.SET_NULL) # OneToOneField: One
9     name = models.CharField(max_length=200, null=True)
10    email = models.CharField(max_length=200, null=True)
11
12    def __str__(self):
13        return self.name
14
```

- app/views.py

ShoppingCart > app > views.py > catalog

```
9
10
11 # Create your views here.
12 def home(request):
13     mongo_data = product_collection.find()
14     products = [parse_product_from_mongo(data) for data in mongo_data]
15
16     # products = Product.objects.all()
17
18     context = get_context({'products': products})
19     return render(request, 'app/home.html', context)
20
21
22 def checkout(request):
23     context = get_context({})
24     return render(request, 'app/checkout.html', context)
25
26
27 def catalog(request, category_slug):
28     query = {'category': category_slug}
29     products_data = product_collection.find(query)
30
31     products = [parse_product_from_mongo(product_data) for product_data in product_data]
32
33     if not products:
34         return HttpResponse(product_collection.find())
35
36     context = get_context({'products': products})
37     return render(request, 'app/catalog.html', context)
38
```

ShoppingCart > app > views.py > catalog

```
40 def search(request):
41     keyword = request.POST.get('keyword')
42     if request.method == 'POST':
43         keyword = request.POST.get('keyword')
44
45     product_collection.create_index([('name', 'text'), ('author', 'text'), ('description', 'text')])
46
47     # Truy vấn để tìm kiếm kể cả khi keyword không đầy đủ
48     search_keyword = re.compile(f".*{re.escape(keyword)}.*", re.IGNORECASE)
49     query = {
50         '$or': [
51             {'name': {'$regex': search_keyword}},
52             {'author': {'$regex': search_keyword}},
53             {'description': {'$regex': search_keyword}}
54         ]
55     }
56     products_data = product_collection.find(query)
57     products = [parse_product_from_mongo(product_data) for product_data in products_data]
58
59     if not products:
60         print(f"No products found with keyword: {keyword}")
61
62     context = get_context({'keyword': keyword, 'products': products})
63     return render(request, 'app/search.html', context)
64
65
66 # BASE
67 def get_base_context():
68     mongo_data = category_collection.find()
69     categories = [parse_category_from_mongo(data) for data in mongo_data]
70     context = {'categories': categories}
71     return context
72
73
74 def get_context(dict2):
75     res = {**get_base_context(), **dict2}
76     return res
77
```

- **app/urls.py**

```
urlpatterns = [
    path('', views.home, name="home"),
    path('catalog/<str:category_slug>/', views.catalog, name="catalog"),
    path('cart/index', cart.viewCart, name="cart"),
    path('cart/changeQuantity', cart.changeQuantity, name='changeQuantity'),
    path('checkout/', views.checkout, name="checkout"),
    path('search/', views.search, name="search"),
]
```

- ShoppingCart
 - INSTALLED_APPS

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app',
    'product',
    'catalog',
    'cart'
]
```

- Database


```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'shoppingcart',  
        'USER': 'root',  
        'PASSWORD': '888888',  
        'HOST': 'localhost'  
    }  
}
```

```
import pymongo
```

```
url = 'mongodb://localhost:27017'  
client = pymongo.MongoClient(url)  
db = client['shoppingcart']
```

- urlpatterns

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('app.urls')),  
    path('product/', include('product.urls')),  
    path('category/', include('catalog.urls')),  
    path('cart/', include('cart.urls'))  
]
```